

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re patent application of:
Andra et al.

Serial No.: 10/709,793

Filed: May 28, 2004

Group Art Unit: 2168

Examiner: Mahesh H. Dwivedi

Atty. Docket No.: BUR920040087US1

For: A SYSTEM AND METHOD FOR SPEEDING XML CONSTRUCTION FOR A
BUSINESS TRANSACTION USING PREBUILT XML WITH STATIC AND
DYNAMIC SECTIONS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' APPEAL BRIEF

Sirs:

Appellants respectfully appeal the final rejection of claims 1-6, 8-16, and 18-24 in the Office Action dated January 2, 2008. A Notice of Appeal was timely filed on March 27, 2008.

Serial No. 10/709,793

Appeal Brief

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, Armonk, New York, assignee of 100% interest of the above-referenced patent application.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative, or Assignee, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-6, 8-16, and 18-24, all the claims pending in the application and set forth fully in the attached appendix (Section IX), are under appeal. Claims 1-22 were originally filed in the application. A final Office Action was issued on January 2, 2008 rejecting claims 1-6, 8-16, and 18-24. The Appellants filed an Amendent under 37 C.F.R. §1.116 on February 28, 2008. An Advisory Action was issued on March 18, 2008 indicating that the Amendent filed under 37 C.F.R. §1.116 on February 28, 2008 would not be entered. The Appellants timely filed a Notice of Appeal on March 27, 2008

Claims 1-6, 8-16, and 18-24 stand rejected under 35 U.S.C. §103(a) as unpatentable over U.S. Patent Application Publication No. 2002/0178103 to Dan et al., hereinafter, Dan, in view of U.S. Patent Application Publication No. 2003/0167446 to Thomas, hereinafter, Thomas, and further in view of U.S. Patent Application Publication No. 2002/0042757 to Albazz et al., hereinafter, Albazz.

Appellants respectfully appeal these rejections based on the following discussion.

IV. STATEMENT OF AMENDMENTS

A final Office Action dated January 2, 2008 stated that all the pending claims 1-6, 8-16, and 18-24 were rejected. The claims in the Appendix (Section IX) are shown in their amended form as of the February 28, 2008 amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The Summary of the claimed subject matter is shown by the following annotated, independent claims, wherein paragraph numbers are designated as, for example, [0001], line numbers correspond to the line numbers of the designated paragraph, and Figures are followed by an element number, where appropriate:

Independent Claim 1

A method for constructing extensible markup language (XML) transactions comprising an XML format run on a computer system, said method comprising:

establishing an original pre-defined data type definition format for an XML transaction ([0025], lines **12-15**; [0026], lines **2-5**; [0028], lines **1-5**; [0029], lines **3-4**);

creating a copy of said original pre-defined data type definition format for said XML transaction ([0025], lines **3-4**; [0028], lines **5-6**);

pre-building static structures of said XML transaction ([0024], line **4**; Fig. 1, 101), wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile ([0024], lines **13-15**; [0026], lines **4-5**; [0027], lines **1-3**; [0028], lines **6-7**; [0029], lines **4-6**; Fig. 2);

classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures ([0024], lines **4-6**; Fig. 1, 103; [0026], lines **7-9**; [0027], lines **3-4**);

Appeal Brief

building a list of a sequence of said static and dynamic structures ([0024], lines **6-7**; Fig. 1, 105; [0027], lines **5-6**);

linking said list to the XML transaction and said predetermined trading partner profile ([0024], lines **7-8**; Fig. 1, 107; [0029], lines **6-7**);

combining said static structures with said dynamic structures at a runtime to construct said XML transaction based on said sequence, said XML transaction, said trading partner profile, and said dynamic structures of said XML transaction ([0024], lines **8-10**; Fig. 1, 109; Figs. 2 and 3), wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner ([0028], lines **8-13**), wherein said combining comprises filling the empty tags of said dynamic structures ([0025], lines **7-9**); and

constructing a final XML structure based on the combining process ([0025], line **5**; Figs. 2 and 3), wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags ([0026], lines **5-7**), and wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction ([0029], lines **9-10**).

Independent Claim 11

A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method for constructing extensible markup language (XML) transactions comprising an XML format run on a computer system ([0031], lines **3-7**), said method comprising:

establishing an original pre-defined data type definition format for an XML transaction ([0025], lines **12-15**; [0026], lines **2-5**; [0028], lines **1-5**; [0029], lines **3-4**);

creating a copy of said original pre-defined data type definition format for said XML transaction ([0025], lines **3-4**; [0028], lines **5-6**);

pre-building static structures of said XML transaction ([0024], line **4**; Fig. 1, 101),

Appeal Brief

wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile ([0024], lines **13-15**; [0026], lines **4-5**; [0027], lines **1-3**; [0028], lines **6-7**; [0029], lines **4-6**; Fig. 2);

classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures ([0024], lines **6-7**; Fig. 1, 105; [0027], lines **3-4**);

building a list of a sequence of said static and dynamic structures ([0024], lines **6-7**; Fig. 1, 105; [0027], lines **5-6**);

linking said list to the XML transaction and said predetermined trading partner profile ([0024], lines **7-8**; Fig. 1, 107; [0029], lines **6-7**);

combining said static structures with said dynamic structures at a runtime to construct said XML transaction based on said sequence, said XML transaction, said trading partner profile, and said dynamic structures of said XML transaction ([0024], lines **8-10**; Fig. 1, 109; Figs. 2 and 3), wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner ([0028], lines **8-13**), wherein said combining comprises filling the empty tags of said dynamic structures ([0025], lines **7-9**); and

constructing a final XML structure based on the combining process ([0025], line **5**; Figs. 2 and 3), wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags ([0026], lines **5-7**), and wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction ([0029], lines **9-10**).

Independent Claim 21

A computer system operable for constructing extensible markup language (XML) transactions comprising an XML format, said computer system comprising:

Appeal Brief

means for establishing an original pre-defined data type definition format for an XML transaction ([0032], line 2; [0025], lines 12-15; [0026], lines 2-5; [0028], lines 1-5; [0029], lines 3-4);

means for creating a copy of said original pre-defined data type definition format for said XML transaction ([0033], lines 2-3; [0025], lines 3-4; [0028], lines 5-6);

means for pre-building static structures of said XML transaction ([0032], lines 2-3; [0024], line 4; Fig. 1, 101), wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile ([0032], lines 1-2; [0024], lines 13-15; [0026], lines 4-5; [0027], lines 1-3; [0028], lines 6-7; [0029], lines 4-6; Fig. 2);

means for classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures ([0032], lines 5-6; [0024], lines 6-7; Fig. 1, 105; [0027], lines 3-4);

means for building a list of a sequence of said static and dynamic structures ([0032], lines 5-6; [0024], lines 6-7; Fig. 1, 105; [0027], lines 5-6);

means for linking said list to the type of XML transaction and said predetermined trading partner profile ([0032], lines 5-7; [0033], lines 8-9; [0024], lines 7-8; Fig. 1, 107; [0029], lines 6-7);

means for combining said static structures with said dynamic structures at a runtime of said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction ([0032], lines 7-10; [0024], lines 8-10; Fig. 1, 109; Figs. 2 and 3), wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner ([0028], lines 8-13), wherein said combining comprises filling the empty tags of said dynamic structures ([0032], lines 7-10; [0033], lines 6-8; [0025], lines 7-9); and

means for constructing a final XML structure based on the combining process ([0032], lines 7-10; [0033], lines 6-8; [0025], line 5; Figs. 2 and 3), wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags

Appeal Brief

([0026], lines **5-7**), and wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction ([0029], lines **9-10**).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The issues presented for review by the Board of Patents Appeals and Interferences are whether claims 1-6, 8-16, and 18-24 are unpatentable under 35 U.S.C. §103(a) as unpatentable over Dan, in view of Thomas, and further in view of Albazz.

VII. ARGUMENT

A. The Position in the Final Office Action

1. The Rejection of Independent Claims 1, 11, and 21 under 35 U.S.C. §103(a) over Dan, in view of Thomas, and further in view of Albazz

Claims 1-6, 8-16, and 18-24 are rejected under 35 U.S.C. 103(a) as unpatentable over Dan et al. (U.S. PGPUB 2002/0178103), in view of Thomas (U.S. PGPUB 2003/0167446), and in view of Albazz et al. (U.S. PGPUB 2002/00427 57).

Regarding claims 1 and 11, Dan teaches a method and a program storage device comprising:

A) establishing an original pre-defined data type definition format for an XML transaction (Paragraphs 31, 50, & 58, Figures 8-9);

C) pre-building static structures of said XML transaction (Paragraphs 33-35);

E) classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures (Paragraphs 34-35);

I) wherein an occurrence of said runtime of said XML transaction occurs when

Appeal Brief

Said XML transaction occurs when said XML transaction is sent to a trading partner (Paragraphs 33-34, 36);

J) wherein said combining comprises filling the empty tags of said dynamic structures (Paragraphs 34-35); and

K) constructing a final XML structure based on the combining process (Paragraph 46);

L) wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags (Paragraphs 34-35, 46).

The examiner notes that Dan teaches “establishing an original pre- defined data type definition format for an XML transaction” as “According to the invention, a meta-contract governs or controls the negotiation process. The meta contract is either pre-negotiated or formed from information provided by the parties in one or more electronic documents, preferably in the form of profiles, described in greater detail below. . . Before creating a meta-contract, the parties must first accept a negotiation protocol to be used during the negotiation process. After the parties accept the negotiation protocol, a meta-contract may be formed and the parties may begin a negotiation” (Paragraph 31), “FIG. 8 illustrates the preferred data type definition (DTD) covering all offer documents” (Paragraph 50), and “FIG. 9 illustrates the preferred data type definition (DTD) covering all counter offer documents” (Paragraph 58). The examiner further notes that Dan teaches “pre-building static structures of said XML transaction” as “The profile serves as the starting point of a negotiation by providing an initial version of a contract document” (Paragraph 33), “The profile may be expressed, for example, as an XML document whose contents may be incorporated into a contract” (Paragraph 34), and “One example of a contract template is an almost-complete electronic contract document with a few fields left blank” (Paragraph 34). The examiner further notes that Dan teaches “classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures” as “a negotiable field 1023 or 1024 may be treated as a blank that me be completed by the negotiating party”

Appeal Brief

(Paragraph 35). The examiner further notes that Dan teaches “wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction occurs when said XML transaction is sent to a trading partner” as “The profile serves as the starting point of a negotiation by providing an initial version of a contract document” (Paragraph 33), “The profile may be expressed, for example, as an XML document whose contents may be incorporated into a contract” (Paragraph 34), and “One example of a contract template is an almost-complete electronic contract document with a few fields left blank” (Paragraph 34). The examiner further wishes to state that the initial contract must combine the static fields (almost complete portions) and dynamic fields (the blank portions) at runtime (when the contract is sent to other party). The examiner further notes that Dan teaches “wherein said combining comprises filling the empty tags of said dynamic structures” as “a negotiable field I 023 or I 024 may be treated as a blank that me be completed by the negotiating party” (Paragraph 35) The examiner further notes that Dan teaches “constructing a final XML structure based on the combining process” as “the negotiation continues 370 to step 380 where the negotiation is complete and step 390 leads to the service contract or TPA” (Paragraph 46). The examiner further notes that Dan teaches “wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags” as “the negotiation continues 370 to step 380 where the negotiation is complete and step 390 leads to the service contract or TPA” (Paragraph 46).

Dan does not explicitly teach:

B) creating a copy of said original pre-defined data type definition format for said XML transaction; and

M) wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction.

Thomas, however, teaches “creating a copy of said original pre-defined data type definition format for said XML transaction” as “the processor reads 12 the document type

Appeal Brief

definition (DTD) of the first XML file and creates a copy 13 of the DTD” (Paragraph 38) and “wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction” as “Once the user has finished entering modifications to the XML file and all of the modifications have been found to be either not significant or valid semantic changes, the temporary version of the XML file in the RAM 7 is written over the original XML file in the first storage region 4” (Paragraph 44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching Thomas’s would have allowed Dan’s to provide a method to record changes to a markup language file by validating them in order to allow that file to be in compliance with constraints defined in a set of declarations, as noted by Thomas (Paragraph 5).

Dan and Thomas do not explicitly teach:

D) wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type;

F) building a list of a sequence of said static and dynamic structures;

G) linking said list to the XML transaction and said predetermined trading partner profile;

H) combining said static structures with said dynamic structures at a runtime to construct said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction.

Albazz, however, teaches “wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type” as “the preferred embodiment of the invention provides for the creation of many different Ts&Cs Sets using the Business Rules Book. Each Ts&Cs Set represents an integrated set of terms and conditions which can be used selectively by the sales group to prepare and propose contracts to prospective buyer organizations. In a marketplace, different Ts&Cs Sets created by a supplier can be used by the e-commerce site to respond to a request for

Appeal Brief

quotation (RFQ) from a buyer either by automatic rating and matching of the request or by pre- assigning a Ts&Cs Set to the buyer” (Paragraph 55) and “During the contract negotiation process the seller may decide to switch into a more attractive Ts&Cs, Set, to overcome buyer reluctance or a competitive offer and win the buyer’s business. This is readily done by simply referencing a different Ts&Cs Set identifier or reference number in the proposed contract or in response to an RFQ. Once a contract is approved and signed by the buyer, a copy of the selected Ts&Cs Set becomes an integral part of that contract. A contract may only include one Ts&Cs Set” (Paragraph 68), “building a list of a sequence of said static and dynamic structures” as “Product List Filter (PLF) is a representation of a (seller’s product list which replaces the complete list of all products available from a seller organization (as used herein the term “products” includes both products and services). This representation comprises products selection and/or exclusion criteria, based on a selection metaphor. The representation criteria are structured and stored in a way that ensures rebuilding the targeted product list from a master product catalog, or from multiple catalogs or other product information sources, any time the target product list is required. Depending upon the used PLF, a generated list could be static with the same products being produced at every run, or could be dynamic with new products being added or removed according to changes taking place at the seller organization. FIG. 5 illustrates an example of the creation and storage of a Product List Filter” (Paragraph 76), “linking said list to the XML transaction and said predetermined trading partner profile” as “PLFs can be implemented within the contract preparation and negotiation cycles in different scenarios. For example, a seller may define a product list to be offered to a particular buyer and create a specific PLF for that list” (Paragraph 79), and “seller can define one or more PLFs that can be linked to offered Ts&Cs Sets or restricted to certain buyers, thus controlling the content of the product list on a buyer-specific basis. The specified buyer(s) become a target buyer for the filtered product list, and PLFs enforce the products viewable by any particular buyer in the execution aspect of the invention, discussed below, whenever the buyer accesses the seller’s e-commerce

Appeal Brief

site (store or marketplace). The buyer can then select or search for required products from the filtered version of the seller's master product list" (Paragraph 80), and "combining said static structures with said dynamic structures at a runtime to construct said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction" as "All contract Static Elements and Dynamic Elements are tied together in a contract profile, which includes linking the Product List Filter(s) and any Dynamic Elements in the Terms and Conditions Set. FIG. 6 illustrates an example of linking a Ts&Cs Page having a multiple Folds to a multiple-tier PLF. Other scenarios might involve linking Ts&Cs Page Folds to other contract elements, for example to different divisions of a buyer organization" (Paragraph 81).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching Albazz's would have allowed Dan's and Thomas's to provide a method to increase the flexibility of contract negotiation through the removal of rigid pre-defined terms and subsequent replacement of dynamic best-fit terms, as noted by Albazz (Paragraph 12).

Regarding claim 21, Dan teaches a computer system comprising:

A) means for establishing an original pre-defined data type definition format for an XML transaction (Paragraphs 31, 50, & 58, Figures 8-9);

C) means for pre-building static structures of said XML transaction (Paragraphs 33-35);

E) means for classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures (Paragraphs 34-35);

I) wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction occurs when said XML transaction is sent to a trading partner (Paragraphs 33-34, 36);

Appeal Brief

J) wherein said combining comprises filling the empty tags of said dynamic structures (Paragraphs 34-35); and

K) means for constructing a final XML structure based on the combining process (Paragraph 46);

L) wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags (Paragraphs 34-35, 46).

The examiner notes that Dan teaches “means for establishing an original pre-defined data type definition format for an XML transaction” as “According to the invention, a meta-contract governs or controls the negotiation process. The meta contract is either pre-negotiated or formed from information provided by the parties in one or more electronic documents, preferably in the form of profiles, described in greater detail below. . . . Before creating a meta- contract, the parties must first accept a negotiation protocol to be used during the negotiation process. After the parties accept the negotiation protocol, a meta- contract may be formed and the parties may begin a negotiation” (Paragraph 31), “FIG. 8 illustrates the preferred data type definition (DTD) covering all offer documents” (Paragraph 50), and “FIG. 9 illustrates the preferred data type definition (DTD) covering all counter offer documents” (Paragraph 58). The examiner further notes that Dan teaches “means for pre-building static structures of said XML transaction” as “The profile serves as the starting point of a negotiation by providing an initial version of a contract document” (Paragraph 33), “The profile may be expressed, for example, as an XML document whose contents may be incorporated into a contract” (Paragraph 34), and “One example of a contract template is an almost-complete electronic contract document with a few fields left blank” (Paragraph 34). The examiner further notes that Dan teaches “means for classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures” as “a negotiable field 1023 or 1024 may be treated as a blank that me be completed by the negotiating party” (Paragraph 35). The examiner further notes that Dan teaches “wherein an occurrence of

Appeal Brief

said runtime of said XML transaction occurs when said XML transaction occurs when said XML transaction is sent to a trading partner” as “The profile serves as the starting point of a negotiation by providing an initial version of a contract document” (Paragraph 33), “The profile may be expressed, for example, as an XML document whose contents may be incorporated into a contract” (Paragraph 34), and “One example of a contract template is an almost- complete electronic contract document with a few fields left blank” (Paragraph 34). The examiner further wishes to state that the initial contract must combine the static fields (almost complete portions) and dynamic fields (the blank portions) at runtime (when the contract is sent to other party). The examiner further notes that Dan teaches “wherein said combining comprises filling the empty tags of said dynamic structures” as “a negotiable field 1023 or 1024 may be treated as a blank that may be completed by the negotiating party” (Paragraph 35) The examiner further notes that Dan teaches “means for constructing a final XML structure based on the combining process” as “the negotiation continues 370 to step 380 where the negotiation is complete and step 390 leads to the service contract or TPA” (Paragraph 46). The examiner further notes that Dan teaches “wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags” as “the negotiation continues 370 to step 380 where the negotiation is complete and step 390 leads to the service contract or TPA” (Paragraph 46).

Dan does not explicitly teach:

B) means for creating a copy of said original pre-defined data type definition format for said XML transaction; and

M) wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction.

Thomas, however, teaches “means for creating a copy of said original pre-defined data type definition format for said XML transaction” as “the processor reads 12 the document type definition (DTD) of the first XML file and creates a copy 13 of the DTD”

Appeal Brief

(Paragraph 38) and “wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction” as “Once the user has finished entering modifications to the XML file and all of the modifications have been found to be either not significant or valid semantic changes, the temporary version of the XML file in the RAM 7 is written over the original XML file in the first storage region 4” (Paragraph 44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching Thomas’s would have allowed Dan’s to provide a method to record changes to a markup language file by validating them in order to allow that file to be in compliance with constraints defined in a set of declarations, as noted by Thomas (Paragraph 5).

Dan and Thomas do not explicitly teach:

D) wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type;

F) means for building a list of a sequence of said static and dynamic structures;

G) means for linking said list to the type of XML transaction and said predetermined trading partner profile;

H) means for combining said static structures with said dynamic structures at a runtime of said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction.

Albazz, however, teaches “wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type” as “the preferred embodiment of the invention provides for the creation of many different Ts&Cs Sets using the Business Rules Book. Each Ts&Cs Set represents an integrated set of terms and conditions which can be used selectively by the sales group to prepare and propose contracts to prospective buyer organizations. In a marketplace, different Ts&Cs Sets created by a supplier can be used by the e-commerce site to respond to a request for quotation (RFQ) from a buyer either by automatic rating and matching of the request or

Appeal Brief

by pre-assigning a Ts&Cs Set to the buyer” (Paragraph 55) and “During the contract negotiation process the seller may decide to switch into a more attractive Ts&Cs Set, to overcome buyer reluctance or a competitive offer and win the buyer’s business. This is readily done by simply referencing a different Ts&Cs Set identifier or reference number in the proposed contract or in response to an RFQ. Once a contract is approved and signed by the buyer, a copy of the selected Ts&Cs Set becomes an integral part of that contract. A contract may only include one Ts&Cs Set” (Paragraph 68), “means for building a list of a sequence of said static and dynamic structures” as “Product List Filter (PLF) is a representation of a seller’s product list which replaces the complete list of all products available from a seller organization (as used herein the term “products” includes both products and services). This representation comprises products selection and/or exclusion criteria, based on a selection metaphor. The representation criteria are structured and stored in a way that ensures rebuilding the targeted product list from a master product catalog, or from multiple catalogs or other product information sources, any time the target product list is required. Depending upon the used PLF, a generated list could be static with the same products being produced at every run, or could be dynamic with new products being added or removed according to changes taking place at the seller organization. FIG. 5 illustrates an example of the creation and storage of a Product List Filter” (Paragraph 76), “means for linking said list to the type of XML transaction and said predetermined trading partner profile” as “PLFs can be implemented within the contract preparation and negotiation cycles in different scenarios. For example, a seller may define a product list to be offered to a particular buyer and create a specific PLF for that list” (Paragraph 79), and “seller can define one or more PLFs that can be linked to offered Ts&Cs Sets or restricted to certain buyers, thus controlling the content of the product list on a buyer-specific basis. The specified buyer(s) become a target buyer for the filtered product list, and PLFs enforce the products viewable by any particular buyer in the execution aspect of the invention, discussed below, whenever the buyer accesses the seller’s e-commerce site (store or marketplace). The buyer can then select or search

Appeal Brief

for required products from the filtered version of the seller's master product list" (Paragraph 80), and "means for combining said static structures with said dynamic structures at a runtime of said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction" as "All contract Static Elements and Dynamic Elements are tied together in a contract profile, which includes linking the Product List Filter(s) and any Dynamic Elements in the Terms and Conditions Set. FIG. 6 illustrates an example of linking a Ts&Cs Page having a multiple Folds to a multiple-tier PLF. Other scenarios might involve linking Ts&Cs Page Folds to other contract elements, for example to different divisions of a buyer organization" (Paragraph 81).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching Albazz's would have allowed Dan's and Thomas's to provide a method to increase the flexibility of contract negotiation through the removal of rigid pre-defined terms and subsequent replacement of dynamic best-fit terms, as noted by Albazz (Paragraph 12).

B. The Appellants' Position

1. Response to the Rejection of Independent Claims 1, 11, and 21 under 35 U.S.C. §103(a) over Dan, in view of Thomas, and further in view of Albazz

a. The Dan Disclosure

[0033] The TPA template or party profile may be included as part of the information advertised by the service provider in step 60 of FIG. 2. The profile serves as the starting point of a negotiation by providing an initial version of a contract document. The profile may include information such as: products and services provided, specific business processes that the service provider can perform, security requirements, and technology information such as which message-exchange protocols are supported by the service provider. The service provider's profile may be embodied in a variety of different forms. Several examples of the service provider's profile are described herein, although alternative profile forms will be apparent to those of ordinary skill in the art. (cited by the Office Action).

[0034] In one embodiment, the service provider's profile may describe the capabilities of one party. This profile may be expressed, for example, as an XML document whose contents may be incorporated into a contract. The information contained in the profile may include not only the capabilities of a party but also may contain requirements of the interacting party in the form of a contract template. The contract template is provided to express a contract either between a pair of roles or between an actual party (whose profile is represented by the template) and a role. One example of a contract template is an almost-complete electronic contract document with a few fields left blank: these fields are to be filled in by the negotiating party. An enhanced template additionally specifies, in an associated document, the acceptable choices for the negotiable fields. (cited by the Office Action).

[0035] Fig. 4 is a schematic diagram that illustrates a party profile and a contract template. Party profile 1010 may contain, for example, party contact information 1011, a description of the service offered or needed 1012, one or more contract templates 1013, and allowable choices 1014. Allowable choices 1014 may cover, for example, business and/or technical considerations such as a list of supported transport protocols, a list of supported shipping or transport services (such as overnight shipping, airmail delivery, etc.), delivery times, and/or the optional use of a preexisting meta contract. Profile 1010 may include a contract template 1020 containing one or more nonnegotiable fields 1021, 1022 and one or more negotiable fields 1023, 1024. As mentioned above, negotiable field 1023 or 1024 may be treated as a blank that may be completed by the negotiating party or, alternatively, may specify capabilities or allowable choices that may be selected. The capabilities and/or allowable choices may be provided as searchable information by a public registry or repository. (cited by the Office Action).

[0046] If either of the parties chooses to abort 340 the negotiation at any time during negotiations, the outcome 350 is the termination of the entire negotiation process. When all the sub components of a contract have been agreed to by both sides, the negotiation is committed in step 360, and the negotiation continues to step 380 where the negotiation is complete and step 380 leads to the service contract or TPA. (cited by the Office Action).

b. The Thomas Disclosure

[0044] Once the user has finished entering modifications to the XML file and all of the modifications have been found to be either not significant or valid semantic changes, the temporary version of the XML file in the RAM 7 is written over the original file in the first storage

region. Of course, the modified version of the XML file may be stored separately from the original version of the XML file instead of overwriting the original XML version. (cited by the Office Action).

c. The Albazz Disclosure

[0076] A product List Filter (PLF) is a representation of a seller's product list which replaces the complete list of all products available from a seller organization (as used herein the term "products" includes both products and services). This representation comprises product selection and/or exclusion criteria, based on a selection metaphor. The representation criteria are structured and stored in a way that ensures rebuilding the targeted product list from a master product catalog, or from multiple catalogs or other product information sources, any time the target product list is required. Depending upon the used PLF, a generated list could be static with the same products being produced at every run, or could be dynamic with new products being added or removed according to changes taking place at the seller organization. Fig. 5 illustrates an example of the creation and storage of a Product List Filter. (cited by the Office Action).

d. Appellants' Arguments

Regarding the rejection of independent claims 1, 11, and 21, the Final Office Action cites Dan for teaching "pre-building static structures of said XML transaction (Paragraphs 33-35)", (Final Action, page 3, section 8., C)).

Dan discloses a contract template 1011, presumably analogous to a business transaction, containing one or more nonnegotiable fields 1021, 1022 and one or more negotiable fields 1023, 1024. Dan also discloses that the profile serves as the starting point of a negotiation by providing an initial version of a contract document, i.e., the contract template, 1011. (Paragraph [0035]).

The present invention defines a "static structure" as "a pre-built XML structure with pre-filled values based on the associated transaction type and TPP [Trading Partner Profile]", (Specification, paragraph [0024], lines 13-15). Therefore, the pre-built XML "static structures" of the present invention are static, i.e., unchanging, and pre-filled with values based on the associated transaction type and trading partner profile. Hence, there

Appeal Brief

are no negotiable fields in the static structures with pre-filled values of the present invention because the structures are static and pre-filled.

In contrast, the contact template of Dan contains one or more negotiable fields 1023, 1024 that will be filled with future negotiations.

Therefore, Applicants respectfully submit that Dan does not disclose, teach or suggest the present invention's feature of "pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile", as recited in independent claims 1, 11, and 21.

Regarding the rejection of independent claims 1, 11, and 21, the Final Office Action cites Thomas for teaching "wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction" as "Once the user has finished entering modifications to the XML file and all of the modifications have been found to be either not significant or valid semantic changes, the temporary version of the XML file in the RAM 7 is written over the original XML file in the first storage region 4" (Paragraph 44 [of Thomas]). (Final Office Action, page 5, line 27 to page 6, line 3).

In the present invention, a business partner agrees to send a business transaction in a mutually agreed upon XML format (proprietary or standards based), call a Data Type Definition (DTD) format. Next, a Trading Partner Profile (TPP) is created in a database that holds information about the partner, the communication protocol used, the enabled transaction, format of transaction, and XML format version. Then, a copy of the DTD is created. Thereafter, static elements of the XML are filled with pre-determined values based on the TPP and are stored using an editor. The static sections are linked to the TPP and transaction, and are stored in the database. At execution time, based on the TPP and transaction combination, the corresponding static sections are taken and the application specific dynamic sections are built to construct the final XML. Execution time is defined

Appeal Brief

as the transaction runtime (that is, sending the transaction to the trading partner). The constructed XML is then run against (compared against) the DTD to validate the structure. (Specification, Paragraph [0028]). (emphases added).

Thomas discloses that the temporary copy of the contents of the XML file is displayed 29 by means of the output interface 10 so that a user is able to input modifications to the XML file via the input interface 9. Each of the changes entered by the user is compared 30 to the temporary copy of the XML file and checked 31 to establish whether the modification is significant, i.e., a semantic change. ... Where the modification is identified as a semantic change, the processor checks 34 whether a valid difference representation of the change can be generated using the delta DTD [i.e., Document Type Definition]. (Paragraph [0043], lines 2-8 and 13-16).

Thomas further discloses that the DTD of the delta file and the DTD of the original markup language file are substantially identical in that the nested structure of the contents of the delta file and the original file will be substantially the same. However, certain slight modifications are necessary to enable the delta file to represent the changes to the markup language file. All of the element type declarations are either copied across to the delta DTD or amended and the element type declarations are processed to the lowest level of each content particle. (Paragraph [0048], lines 7-15).

Thomas yet further discloses a method of recording and validating changes to a markup language, wherein semantic changes to the original XML file necessarily require slight modifications to enable a delta file to represent changes to the original markup language file. These slight modifications may entail amending element type declarations and processing these element type declarations to the lowest level of each content particle.

However, the DTD of the present invention is originally fixed and its DTD copied, and does not require such slight modifications or amending of element type declarations as does Thomas because the final XML structure of the present invention comprises pre-filled static structures, to which no modifications or amendments of the DTD are made, and dynamic structures that comprise empty tags, to which no modifications or amendments of

Appeal Brief

the DTD are made, so that the final or constructed XML structure may be compared to or validated against the original pre-defined data type definition.

Therefore, Applicants respectfully submit that Thomas does not disclose, teach or suggest the present invention's feature of "wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction", as recited in independent claims 1, 11, and 21.

Furthermore, for at least the reasons outlined immediately above, Applicants respectfully submit that Thomas does not cure the deficiencies of Dan, because Thomas does not disclose, teach or suggest the present invention's feature of "pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile".

Instead, Thomas merely discloses a method of recording and validating changes to a markup language, wherein semantic changes to the original XML file necessarily require slight modifications to enable a delta file to represent changes to the original markup language file.

Regarding the rejection of independent claims 1, 11, and 21, the Final Office Action cites Albazz for teaching "building a list of a sequence of said static and dynamic structures". (Final Office Action, page 7, lines 6 and 7).

Albazz merely discloses generating a list that could be static with the same products being produced at every run, or could be dynamic with new products being added or removed according to changes taking place at the seller organization. (Paragraph [0076]).

In contrast, the present invention describes the feature of "building a list of a sequence of said static and dynamic structures", wherein both static and dynamic structures are previously defined, i.e., "pre-building static structures of said XML transaction,

Appeal Brief

wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile; classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures".

The static and dynamic product lists of Albazz do not disclose, teach or suggest the present invention's features of "wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile" or "dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures". The static and dynamic product lists of Albazz are not explicitly defined as XML data structures corresponding to a transaction. A list is not a transaction.

Therefore, Applicants respectfully submit that Albazz does not disclose, teach or suggest the present invention's feature of "pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile; classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures; building a list of a sequence of said static and dynamic structures", as recited in independent claims 1, 11, and 21.

Furthermore, for at least the reasons outlined immediately above with respect to Albazz and above with respect to Dan and Thomas, Applicants respectfully submit that Albazz does not cure the deficiencies of Dan and Thomas, because Albazz also does not disclose, teach or suggest the present invention's feature of "pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile". Instead, merely discloses generating a list that could be static with the same products being produced at every run, or could be dynamic

Appeal Brief

with new products being added or removed according to changes taking place at the seller organization.

For at least the reasons outlined above, Applicants respectfully submit that Dan, Thomas, and Albazz, either individually or in combination, do not disclose, teach or suggest the present invention's features of "pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile; classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures; building a list of a sequence of said static and dynamic structures", as recited in independent claims 1, 11, and 21. Accordingly, Dan, Thomas, and Albazz, either individually or in combination, fail to render obvious the subject matter of independent claims 1, 11, and 21, and dependent claims 2-6, 8-10, 12-20, and 21-24 under 35 U.S.C. §103(a).

In view of the foregoing, the Board is respectfully requested to reconsider and withdraw the rejection of independent claims 1, 11, and 21 and dependent claims 2-6, 8-10, 12-20, and 21-24 under 35 U.S.C. §103(a) as unpatentable over Dan, Thomas, and Albazz

VIII. CONCLUSION

In view of the foregoing, the Appellants respectfully submit that the collective cited prior art do not teach or suggest the features defined by independent claims 1, 11 and 21, and as such, claims 1, 11, and 21 are patentable over Dan, Thomas and Alabazz, either individually or in combination. Furthermore, as the independent claims are patentable over Dan, Thomas, and Albazz, so too are dependent claims 2-6, 8-10, 12-16, 18-20, and 22-24 patentable over Dan, Thomas and Albazz by virtue of their dependency from patentable independent claims 1, 11, and 21. Thus, Appellants respectfully request that the Board reconsider and withdraw the rejections of claims 1-6, 8-16, and 18-24 and pass these claims to issue.

Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0456.

Respectfully submitted,

Date: May 27, 2008

/Peter A. Balnave/
Peter A. Balnave, Ph.D.
Registration No. 46,199

Gibb & Rahman, LLC
2568-A Riva Road, Suite 304
Annapolis, MD, 21401
Voice: (410) 573-5255
Fax: (301) 261-8825
Balnave@Gibb-Rahman.com
Customer No. 29154

IX. CLAIMS APPENDIX

1. A method for constructing extensible markup language (XML) transactions comprising an XML format run on a computer system, said method comprising:
 - establishing an original pre-defined data type definition format for an XML transaction;
 - creating a copy of said original pre-defined data type definition format for said XML transaction;
 - pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile;
 - classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures;
 - building a list of a sequence of said static and dynamic structures;
 - linking said list to the XML transaction and said predetermined trading partner profile;
 - combining said static structures with said dynamic structures at a runtime to construct said XML transaction based on said sequence, said XML transaction, said trading partner profile, and said dynamic structures of said XML transaction, wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner, wherein said combining comprises filling the empty tags of said dynamic structures; and
 - constructing a final XML structure based on the combining process, wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags, and wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction.

Appeal Brief

2. The method of claim 1, wherein said XML transaction occurs in a business-to-business (B2B) electronic environment.
3. The method of claim 1, further comprising predefining said trading partner profile associated with a predetermined trading entity.
4. The method of claim 1, wherein said pre-building of said static structures occurs prior to runtime of said XML transaction.
5. The method of claim 1, wherein the construction of said final XML structure follows definitions established by said copy of said original pre-defined data type definition format for said XML transaction.
6. The method of claim 1, further comprising filling said empty tags of said dynamic structures with business data values and building multiple repeating dynamic structures at runtime of said XML transaction.
7. (Cancelled).
8. The method of claim 3, wherein said trading partner profile comprises partner data, communication protocol data, transaction data, transaction format data, and XML format version data.
9. The method of claim 3, wherein said pre-building of said static structures and a pre-building of said dynamic structures occurs at a time of installation of said trading partner profile in a database in said computer system.
10. The method of claim 9, further comprising:

Appeal Brief

linking said static structures to a type of XML transaction and said predetermined trading partner profile; and
storing the linked static structures in said database.

11. A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method for constructing extensible markup language (XML) transactions comprising an XML format run on a computer system, said method comprising:

establishing an original pre-defined data type definition format for an XML transaction;

creating a copy of said original pre-defined data type definition format for said XML transaction;

pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile;

classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures;

building a list of a sequence of said static and dynamic structures;

linking said list to the type of XML transaction and said predetermined trading partner profile;

combining said static structures with said dynamic structures at a runtime of said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction, wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner, wherein said combining comprises filling the empty tags of said dynamic structures; and

constructing a final XML structure based on the combining process, wherein said final XML structure comprises fully built dynamic structures that comprise completely

Appeal Brief

filled tags, and wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction.

12. The program storage device of claim 11, wherein said XML transaction occurs in a business-to-business (B2B) electronic environment.

13. The program storage device of claim 11, wherein said method further comprises predefining said trading partner profile associated with a predetermined trading entity.

14. The program storage device of claim 11, wherein said pre-building of said static structures occurs prior to runtime of said XML transaction.

15. The program storage device of claim 11, wherein the construction of said final XML structure follows definitions established by said copy of said original pre-defined data type definition format for said XML transaction.

16. The program storage device of claim 11, wherein said method further comprises filling said empty tags of said dynamic structures with business data values and building multiple repeating dynamic structures at runtime of said XML transaction.

17. (Cancelled).

18. The program storage device of claim 13, wherein said trading partner profile comprises partner data, communication protocol data, transaction data, transaction format data, and XML format version data.

19. The program storage device of claim 13, wherein said pre-building of said static

Appeal Brief

structures and a pre-building of said dynamic structures occurs at a time of installation of said trading partner profile in a database in said computer system.

20. The program storage device of claim 19, wherein said method further comprises:
linking said static structures to a type of XML transaction and said predetermined trading partner profile; and

storing the linked static structures in said database.

21. A computer system operable for constructing extensible markup language (XML) transactions comprising an XML format, said computer system comprising:

means for establishing an original pre-defined data type definition format for an XML transaction;

means for creating a copy of said original pre-defined data type definition format for said XML transaction;

means for pre-building static structures of said XML transaction, wherein said static structures comprise a pre-built XML data structure with pre-filled values based on a transaction type of said XML transaction and a predetermined trading partner profile;

means for classifying dynamic structures of said XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures;

means for building a list of a sequence of said static and dynamic structures;

means for linking said list to the type of XML transaction and said predetermined trading partner profile;

means for combining said static structures with said dynamic structures at a runtime of said XML transaction based on said sequence, said type of XML transaction, said trading partner profile, and said dynamic structures of said XML transaction, wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner, wherein said combining comprises filling the empty tags of said dynamic structures; and

Appeal Brief

means for constructing a final XML structure based on the combining process, wherein said final XML structure comprises fully built dynamic structures that comprise completely filled tags, and wherein said final XML structure is validated by comparing said final XML structure against said copy of said original pre-defined data type definition format for said XML transaction.

22. The computer system of claim 21, further comprising:

means for predefining said trading partner profile associated with a predetermined trading entity;

means for filling said empty tags of said dynamic structures with business data values and building multiple repeating dynamic structures at runtime of said XML transaction;

means for linking said static structures to a type of XML transaction and said predetermined trading partner profile; and

means for storing the linked static structures.

23. The computer system of claim 21, wherein said static structures are pre-built prior to runtime of said XML transaction.

24. The computer system of claim 21, wherein said static structures and said dynamic structures are pre-built at a time of installation of said trading partner profile in a database of said computer system.

X. EVIDENCE APPENDIX

There is no other evidence known to Appellants, Appellants' legal representative, or Assignee, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There is no other related proceeding known to Appellants, Appellants' legal representative, or Assignee, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.